09/869435
PCT/FR00/02978

INPI
INSTITUT NATIONAL DE LA PROPRIETE INDUSTRIELLE

EJC

BREVET D'INVENTION

CERTIFICAT D'UTILITÉ - CERTIFICAT D'ADDITION

COPIE OFFICIELLE

Le Directeur général de l'Institut national de la propriété industrielle certifie que le document ci-annexé est la copie certifiée conforme d'une demande de titre de propriété industrielle déposée à l'Institut.

Fait à Paris, le 09 NOV. 2000

Pour le Directeur général de l'Institut
national de la propriété industrielle
Le Chef du Département des brevets

Martine PLANCHE

DOCUMENT DE PRIORITÉ
PRÉSENTÉ OU TRANSMIS CONFORMÉMENT À LA REGLE 17.1.a) OU b)

INSTITUT NATIONAL DE LA PROPRIETE INDUSTRIELLE
SIEGE
26 bis, rue de Saint Petersbourg
75800 PARIS cedex 08
Téléphone : 01 53 04 53 04
Télécopie : 01 42 93 59 30
http://www.inpi.fr

INSTITUT NATIONAL DE LA PROPRIETE INDUSTRIELLE

26 bis, rue de Saint Pétersbourg
75800 Paris Cedex 08
Téléphone : 01 53 04 53 04 Télécopie : 01 42 93 59 30

BREVET D'INVENTION, CERTIFICAT D'UTILITÉ

Code de la propriété intellectuelle-Livre VI

cerfa N° 55-1328

REQUÊTE EN DÉLIVRANCE

Confirmation d'un dépôt par télécopie

Cet imprimé est à remplir à l'encre noire en lettres capitales

Réservé à l'INPI

DATE DE REMISE DES PIÈCES 28 OCT 1999
N° D'ENREGISTREMENT NATIONAL 9913507
DÉPARTEMENT DE DÉPÔT 75 INPI PARIS
DATE DE DÉPÔT 28 OCT. 1999

1 NOM ET ADRESSE DU DEMANDEUR OU DU MANDATAIRE À QUI LA CORRESPONDANCE DOIT ÊTRE ADRESSÉE

BULL S.A.
Monsieur Bernard CORLU / PC 58D20
68, route de Versailles
78434 LOUVECIENNES CEDEX

n°du pouvoir permanent PG 4280 références du correspondant FR3857/BC téléphone 01 39.66.61.76

2 DEMANDE Nature du titre de propriété industrielle

[X] brevet d'invention [ ] demande divisionnaire
[ ] certificat d'utilité [ ] transformation d'une demande de brevet européen

demande initiale : [ ] brevet d'invention [ ] certificat d'utilité n° date

Établissement du rapport de recherche [ ] différé [X] immédiat

Le demandeur, personne physique, requiert le paiement échelonné de la redevance [ ] oui [X] non

Titre de l'invention (200 caractères maximum)

Procédé de sécurisation d'un ensemble électronique de cryptographie à base d'exponentiation modulaire contre les attaques par analyse physique.

3 DEMANDEUR (S) n° SIREN 3 2 9 5 5 6 1 4 6 code APE-NAF B 3 2 1

Nom et prénoms (souligner le nom patronymique) ou dénomination

BULL CP8

Forme juridique

S.A.

Nationalité (s) Française

Adresse (s) complète (s)

BULL CP8
BP 45
68, route de Versailles
78430 LOUVECIENNES

Pays

FRANCE

En cas d'insuffisance de place, poursuivre sur papier libre

4 INVENTEUR (S) Les inventeurs sont les demandeurs [ ] oui [X] non Si la réponse est non, fournir une désignation séparée

5 RÉDUCTION DU TAUX DES REDEVANCES [ ] requise pour la 1ère fois [ ] requise antérieurement au dépôt : joindre copie de la décision d'admission

6 DÉCLARATION DE PRIORITÉ OU REQUÊTE DU BÉNÉFICE DE LA DATE DE DÉPÔT D'UNE DEMANDE ANTÉRIEURE

pays d'origine	numéro	date de dépôt	nature de la demande

7 DIVISIONS antérieures à la présente demande n° date n° date

8 SIGNATURE DU DEMANDEUR OU DU MANDATAIRE (nom et qualité du signataire)

Bernard CORLU
Mandataire -

SIGNATURE DU PRÉPOSÉ À LA RÉCEPTION SIGNATURE APRÈS ENREGISTREMENT DE LA DEMANDE À L'INPI

INPI
INSTITUT NATIONAL DE LA PROPRIETE INDUSTRIELLE

BREVET D'INVENTION, CERTIFICAT D'UTILITE

DÉSIGNATION DE L'INVENTEUR
(si le demandeur n'est pas l'inventeur ou l'unique inventeur)

DEPARTEMENT DES BREVETS

26bis, rue de Saint-Pétersbourg
75800 Paris Cédex 08
Tél. : 01 53 04 53 04 - Télécopie : 01 42 93 59 30

FR 3857BC

N° D'ENREGISTREMENT NATIONAL

9913507

TITRE DE L'INVENTION :

Procédé de sécurisation d'un ensemble électronique de cryptographie à base d'exponentiation modulaire contre les attaques par analyse physique.

LE(S) SOUSSIGNÉ(S)

BULL S.A.

DÉSIGNE(NT) EN TANT QU'INVENTEUR(S) (indiquer nom, prénoms, adresse et souligner le nom patronymique) :

GOUBIN Louis
3 rue Brown Sequard
75015 PARIS
France

NOTA : A titre exceptionnel, le nom de l'inventeur peut être suivi de celui de la société à laquelle il appartient (société d'appartenance) lorsque celle-ci est différente de la société déposante ou titulaire.

Date et signature (s) du (des) demandeur (s) ou du mandataire

Louveciennes, le 29 octobre 1999

Corlu Bernard (mandataire)

BA 113/140897

PROCEDE DE SECURISATION D'UN ENSEMBLE ELECTRONIQUE DE CRYPTOGRAPHIE A BASE D'EXPONENTIATION MODULAIRE CONTRE LES ATTAQUES PAR ANALYSE PHYSIQUE

La présente invention concerne un procédé de sécurisation d'un ensemble électronique mettant en œuvre un algorithme faisant intervenir une exponentiation modulaire, dans laquelle l'exposant est secret. Plus précisément, le procédé vise à réaliser une version d'un tel algorithme qui ne soit pas vulnérable face à un certain type d'attaques physiques – dites « analyse d'énergie électrique différentielle ou analyse d'énergie électrique différentielle de haut niveau » (*Differential Power Analysis* ou *High-Order Differential Power Analysis*, en langage anglo-saxon, en abrégé DPA ou HO-DPA) - qui cherchent à obtenir des informations sur la clé secrète à partir de l'étude de la consommation électrique de l'ensemble électronique au cours de l'exécution du calcul.

Les algorithmes cryptographiques considérés ici utilisent une clé secrète pour calculer une information de sortie en fonction d'une information d'entrée ; il peut s'agir d'une opération de chiffrement, de déchiffrement ou de signature ou de vérification de signature, ou d'authentification ou de non-répudiation ou d'échange de clé. Ils sont construits de manière à ce qu'un attaquant, connaissant les entrées et les sorties, ne puisse en pratique déduire aucune information sur la clé secrète elle-même.

On s'intéresse donc à une classe plus large que celle traditionnellement désignée par l'expression *algorithmes à clé secrète* ou *algorithmes symétriques*. En particulier, tout ce qui est décrit dans la présente demande de brevet s'applique également aux algorithmes dits *à clé publique* ou *algorithmes asymétriques*, qui comportent en fait deux clés : l'une publique, et l'autre, privée, non divulguée, cette dernière étant celle visée par les attaques décrites ci-dessous.

Les attaques de type Analyse de Puissance Electrique, développées par Paul Kocher et *Cryptographic Research* (Confer document *Introduction to Differential Power Analysis and related Attacks* by Paul Kocher, Joshua Jaffe, and Benjamin Jun,

Cryptography Research, 870 Market St., Suite 1008, San Francisco, CA 94102, édition du document HTML à l'adresse URL : http://www.cryptography.com/dpa/technical/index.html) partent de la constatation qu'en réalité l'attaquant peut acquérir des informations, autres que la simple donnée des entrées et des sorties, lors de l'exécution du calcul, comme par exemple la consommation électrique du microcontrôleur ou le rayonnement électromagnétique émis par le circuit.

L'analyse d'énergie électrique différentielle est une attaque permettant d'obtenir des informations sur la clé secrète contenue dans l'ensemble électronique, en effectuant une analyse statistique des enregistrements de consommation électrique effectués sur un grand nombre de calculs avec cette même clé.

Cette attaque ne nécessite aucune connaissance sur la consommation électrique individuelle de chaque instruction, ni sur la position dans le temps de chacune de ces instructions. Elle s'applique de la même manière si on suppose que l'attaquant connaît des sorties de l'algorithme et les courbes de consommation correspondantes. Elle repose uniquement sur l'hypothèse fondamentale selon laquelle :

Hypothèse fondamentale : Il existe une variable intermédiaire, apparaissant dans le cours du calcul de l'algorithme, telle que la connaissance de quelques bits de clé, en pratique moins de 32 bits, permet de décider si deux entrées, respectivement deux sorties, donnent ou non la même valeur pour cette variable.

Les attaques dites par analyse d'énergie électrique de haut niveau sont une généralisation de l'attaque DPA décrite précédemment. Elles peuvent utiliser plusieurs sources d'information différentes : outre la consommation, elles peuvent mettre en jeu les mesures de rayonnement électromagnétique, de température, etc. et mettre en œuvre des traitements statistiques plus sophistiqués que la simple notion de moyenne, des variables intermédiaires moins élémentaires qu'un simple bit ou un simple octet.

Néanmoins, elles reposent exactement sur la même hypothèse fondamentale que la DPA.

Le procédé, objet de la présente invention, a pour objet la suppression des risques d'attaques DPA ou HO-DPA d'ensembles ou systèmes électroniques de cryptographie à clé secrète ou privée, faisant intervenir une exponentiation modulaire, dans laquelle l'exposant est secret.

Un autre objet de la présente invention est en conséquence une modification du processus de calcul cryptographique mis en œuvre par les systèmes électroniques de cryptographie protégés de manière que l'hypothèse fondamentale précitée ne soit plus vérifiée, à savoir qu'aucune variable intermédiaire ne dépend de la consommation d'un sous-ensemble aisément accessible de la clé secrète ou privée, les attaques de type DPA ou HO-DPA étant ainsi rendues inopérantes.

Premier exemple : l'algorithme RSA

Le RSA est le plus célèbre des algorithmes cryptographiques asymétriques. Il a été développé par Rivest, Shamir et Adleman en 1978. Pour une description plus détaillée de cet algorithme, on pourra utilement se reporter au document ci-après :

- R.L. Rivest, A. Shamir, L.M. Adleman, *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*, Communications of the ACM, 21, n°2, 1978, pp. 120-126,

ou aux documents suivants :

- ISO/IEC 9594-8/ITU-T X.509, *Information Technology - Open Systems Interconnection - The Directory: Authentication Framework*;
- ANSI X9.31-1, *American National Standard, Public-Key Cryptography Using Reversible Algorithms for the Financial Services Industry*, 1993;
- PKCS #1, *RSA Encryption Standard*, version 2, 1998, disponible à l'adresse suivante :

ftp://ftp.rsa.com/pub/pkcs/doc/pkcs-1v2.doc.

L'algorithme RSA utilise un nombre entier n qui est le produit de deux grands nombres premiers p et q, et un nombre entier e, premier avec $ppcm(p-1,q-1)$, et tel que $e \neq \pm 1 \bmod ppcm(p-1,q-1)$. Les entiers n et e constituent la clé publique. Le calcul en clé publique fait appel à la fonction g de Z/nZ dans Z/nZ définie par $g(x)=x^e \bmod n$. Le calcul en clé secrète fait appel à la fonction $g^{-1}(y)=y^d \bmod n$, où d est l'exposant secret (appelé aussi clé secrète, ou privée) défini par $ed \equiv 1 \bmod ppcm(p-1,q-1)$.

Les attaques de type DPA ou HO-DPA font peser une menace sur les mises en œuvre classiques de l'algorithme RSA. En effet , celles-ci utilisent très souvent le principe dit de *square and multiply* en langage anglo-saxon pour effectuer le calcul de $x^d \bmod n$.

Ce principe consiste à écrire la décomposition

$$d = b_{m-1} \cdot 2^{m-1} + b_{m-2} \cdot 2^{m-2} + \ldots + b_1 \cdot 2^1 + b_0 \cdot 2^0$$

de l'exposant secret d en base 2, puis d'effectuer le calcul de la manière suivante :

1. $z \leftarrow 1$;

pour i allant de $m-1$ jusqu'à 0 faire :

2. $z \leftarrow z^2 \bmod n$;
3. si $b_i = 1$ alors $z \leftarrow z \times x \bmod n$.

Dans ce calcul, on constate que parmi les valeurs successives prises par la variable z, les premières ne dépendent que de quelques bits de la clé secrète d. L'hypothèse fondamentale permettant l'attaque DPA est donc réalisée. On peut ainsi deviner par exemple les 10 bits de poids fort de d en s'intéressant aux mesures de consommation sur la partie de l'algorithme correspondant à i allant de $m-1$ à $m-10$. On peut ensuite continuer l'attaque en utilisant les mesures de consommation sur la partie de l'algorithme correspondant à i allant de $m-11$ à $m-20$, ce qui permet de trouver les 10 bits suivants de d, et ainsi de suite. On trouve finalement tous les bits de l'exposant secret d.

Une première méthode de sécurisation, et ses inconvénients

Une méthode classique (proposée par Ronald Rivest en 1995) pour protéger l'algorithme RSA contre les attaques de type DPA consiste à utiliser un principe de " blinding " (camouflage). On utilise le fait que :

$$x^d \bmod n = (x \times r^e)^d \times r^{-1} \bmod n$$

Ainsi le calcul de $y = x^d \bmod n$ se décompose en quatre étapes :

- On utilise un générateur aléatoire pour obtenir une valeur r ;
- On calcule : $u = x \times r^e \bmod n$;
- On calcule : $v = u^d \bmod n$;
- On calcule : $y = v \times r^{-1} \bmod n$.

L'inconvénient de cette méthode est qu'elle oblige, pour chaque calcul, à calculer l'inverse modulaire r^{-1} de la valeur aléatoire r, cette opération étant en général coûteuse en temps (la durée d'un tel calcul est du même ordre que celle d'une exponentiation modulaire telle que $u^d \bmod n$). Par conséquent, cette nouvelle implémentation (protégée contre les attaques DPA) du calcul de $x^d \bmod n$ est environ deux fois plus lente que l'implémentation initiale (non protégée contre les attaques DPA). En d'autres termes, cette protection du RSA contre les attaques DPA accroît le temps de calcul de **100%** environ (en supposant que l'exposant public e est très petit, par exemple $e=3$; si l'exposant e est plus grand, ce temps de calcul est encore plus grand).

Une deuxième méthode : le procédé de la présente invention

Selon l'invention, un procédé de sécurisation d'un ensemble électronique mettant en œuvre un processus de calcul cryptographique faisant intervenir une exponentiation modulaire d'une grandeur (x), ladite exponentiation modulaire utilisant un exposant secret (d), est caractérisé en ce que l'on décompose ledit exposant secret en une

pluralité de k valeurs imprévisibles $(d_1, d_2, \ldots\ldots, d_k)$ dont la somme est égale audit exposant secret.

Avantageusement, lesdites valeurs $(d_1, d_2, \ldots\ldots, d_k)$ sont obtenues de la manière suivante :

a) $(k-1)$ valeurs sont obtenues au moyen d'un générateur aléatoire ;

b) la dernière valeur est obtenue par différence entre l'exposant secret et les $(k-1)$ valeurs.

Avantageusement, le calcul de l'exponentiation modulaire est effectué de la manière suivante :

a) pour chacune desdites k valeurs, on élève la grandeur (x) à un exposant comprenant ladite valeur pour obtenir un résultat, un ensemble de résultats étant ainsi obtenus ;

b) on calcule un produit des résultats obtenus à l'étape a).

Avantageusement, au moins l'une desdites $(k-1)$ valeurs obtenues au moyen d'un générateur aléatoire a une longueur supérieure ou égale à 64 bits.

Des détails et avantages de la présente invention apparaîtront au cours de la description suivante de quelques modes d'exécution préférés mais non limitatifs, en regard de la figure unique annexée, représentant une carte à puce.

Selon l'invention, on utilise le fait que :

$$si\ d = d_1 + d_2,\ alors\ x^d \bmod n = x^{d_1} \times x^{d_2} \bmod n$$

Ainsi le calcul de $y = x^d \bmod n$ se décompose en cinq étapes :

- On utilise un générateur aléatoire pour obtenir une valeur d_1 ;
- On calcule : $d_2 = d - d_1$;
- On calcule : $u = x^{d_1} \bmod n$;

- On calcule : $v = x^{d_2} \bmod n$;
- On calcule : $y = u \times v \bmod n$.

L'avantage est que, de cette manière, il n'y a pas d'inverse modulaire à calculer. En général, le temps de calcul d'une exponentiation modulaire est proportionnel à la taille de l'exposant. Ainsi si on note α le rapport entre la taille de d_1 et la taille de d_2, on se rend compte que le temps total du calcul dans cette nouvelle implémentation (protégée contre les attaques DPA) est environ $(1+\alpha)$ fois le temps de calcul dans l'implémentation initiale (non protégée contre les attaques DPA).

Notons que, pour obtenir une valeur d_1 non prédictible, il est nécessaire que sa taille soit au moins de 64 bits.

Le procédé ainsi décrit rend inopérantes les attaques de type DPA ou HO-DPA décrites précédemment. En effet, pour décider si deux entrées (respectivement deux sorties) de l'algorithme donnent ou non la même valeur pour une variable intermédiaire apparaissant au cours du calcul, il ne suffit plus de connaître les bits de clé mis en jeu. Il faut également connaître la décomposition de la clé secrète d en k valeurs $d_1, d_2, \ldots, d_k$ telles que $d=d_1+d_2+\ldots+d_k$. Si on suppose que cette décomposition est secrète, et qu'au moins une des k valeurs a une taille d'au moins 64 bits, l'attaquant ne peut pas prévoir les valeurs de $d_1, \ldots, d_k$, et donc l'hypothèse fondamentale, qui permettait de mettre en œuvre une attaque de type DPA ou HO-DPA, n'est plus vérifiée.

Exemples :

1. Si n a une longueur de 512 bits, en choisissant de prendre une valeur aléatoire d_1 de 64 bits, on obtient $\alpha=1/8$, ce qui fait que cette protection du RSA contre les attaques DPA accroît le temps de calcul de **12.5 %** environ.
2. Si n a une longueur de 1024 bits, en choisissant de prendre une valeur aléatoire d_1 de 64 bits, on obtient $\alpha=1/16$, ce qui fait que cette protection du RSA contre les attaques DPA accroît le temps de calcul de **6.25%** environ.

Deuxième exemple : l'algorithme de Rabin

Nous considérons ici l'algorithme cryptographique asymétrique développé par Rabin en 1979. Pour une description plus détaillée de cet algorithme, on pourra utilement se reporter au document suivant :

- M.O. Rabin, *Digitized Signatures and Public-Key Functions as Intractable as Factorization*, Technical Report LCS/TR-212, M.I.T. Laboratory for Computer Science, 1979.

L'algorithme de Rabin utilise un nombre entier n qui est le produit de deux grands nombres premiers p et q, vérifiant en outre les deux conditions suivantes :

- p est congru à 3 modulo 8 ;
- q est congru à 7 modulo 8.

Le calcul en clé publique fait appel à la fonction g de Z/nZ dans Z/nZ définie par $g(x)=x^2 \bmod n$. Le calcul en clé secrète fait appel à la fonction $g^{-1}(y)=y^d \bmod n$, où d est l'exposant secret (appelé aussi clé secrète, ou privée) défini par $d=((p-1)(q-1)/4+1)/2$.

La fonction mise en jeu par le calcul en clé secrète étant exactement la même que celle utilisée par l'algorithme RSA, les mêmes attaques DPA ou HO-DPA sont applicables et font peser les mêmes menaces sur l'algorithme de Rabin.

Sécurisation de l'algorithme

Comme la fonction est exactement la même que celle du RSA, le procédé de sécurisation décrit dans le cadre du RSA s'applique de la même manière au cas de l'algorithme de Rabin. L'accroissement du temps de calcul provoqué par l'application de ce procédé est également le même que dans le cas de l'algorithme RSA.

L'invention peut être mise en oeuvre dans tout ensemble électronique effectuant un calcul cryptographique faisant intervenir une exponentiation modulaire, notamment une carte à puce 8 selon la figure unique. La puce inclut des moyens de traitement de

l'information 9, reliés d'un côté à une mémoire non volatile 10 et à une mémoire volatile de travail RAM 11, et reliés d'un autre côté à des moyens 12 pour coopérer avec un dispositif de traitement de l'information. La mémoire non volatile 10 peut comprendre une partie non modifiable ROM et une partie modifiable EPROM, EEPROM, ou constituée de mémoire RAM du type "flash" ou FRAM (cette dernière étant une mémoire RAM ferromagnétique), c'est-à-dire présentant les caractéristiques d'une mémoire EEPROM avec en outre des temps d'accès identiques à ceux d'une RAM classique.

En tant que puce, on pourra notamment utiliser un microprocesseur autoprogrammable à mémoire non volatile, tel que décrit dans le brevet américain n° 4.382.279 au nom de la Demanderesse. Dans une variante, le microprocesseur de la puce est remplacé - ou tout du moins complété - par des circuits logiques implantés dans une puce à semi-conducteurs. En effet, de tels circuits sont aptes à effectuer des calculs, notamment d'authentification et de signature, grâce à de l'électronique câblée, et non microprogrammée. Ils peuvent notamment être de type ASIC (de l'anglais « Application Specific Integrated Circuit »). Avantageusement, la puce sera conçue sous forme monolithique.

REVENDICATIONS

1. Procédé de sécurisation d'un ensemble électronique mettant en œuvre un processus de calcul cryptographique faisant intervenir une exponentiation modulaire d'une grandeur (x), ladite exponentiation modulaire utilisant un exposant secret (d), caractérisé en ce que l'on décompose ledit exposant secret en une pluralité de k valeurs imprévisibles $(d_1, d_2, \ldots\ldots, d_k)$ dont la somme est égale audit exposant secret.

2. Procédé selon la revendication 1, caractérisé en ce que lesdites valeurs $(d_1, d_2, \ldots\ldots, d_k)$ sont obtenues de la manière suivante :

a) $(k-1)$ valeurs sont obtenues au moyen d'un générateur aléatoire ;

b) la dernière valeur est obtenue par différence entre l'exposant secret et les $(k-1)$ valeurs.

3. Procédé selon la revendication 1, caractérisé en ce que le calcul de l'exponentiation modulaire est effectué de la manière suivante :

a) pour chacune desdites k valeurs, on élève la grandeur (x) à un exposant comprenant ladite valeur pour obtenir un résultat, un ensemble de résultats étant ainsi obtenus ;

b) on calcule un produit des résultats obtenus à l'étape a).

4. Procédé selon la revendication 1, caractérisé en ce qu'au moins l'une desdites $(k-1)$ valeurs obtenues au moyen d'un générateur aléatoire a une longueur supérieure ou égale à 64 bits.

5. Utilisation du procédé selon la revendication 1 dans une carte à puce comportant des moyens de traitement de l'information.

6. Utilisation du procédé selon la revendication 1 pour la sécurisation d'un processus de calcul cryptographique utilisant l'algorithme RSA.

7. Utilisation du procédé selon la revendication 1 pour la sécurisation d'un processus de calcul cryptographique utilisant l'algorithme de Rabin.

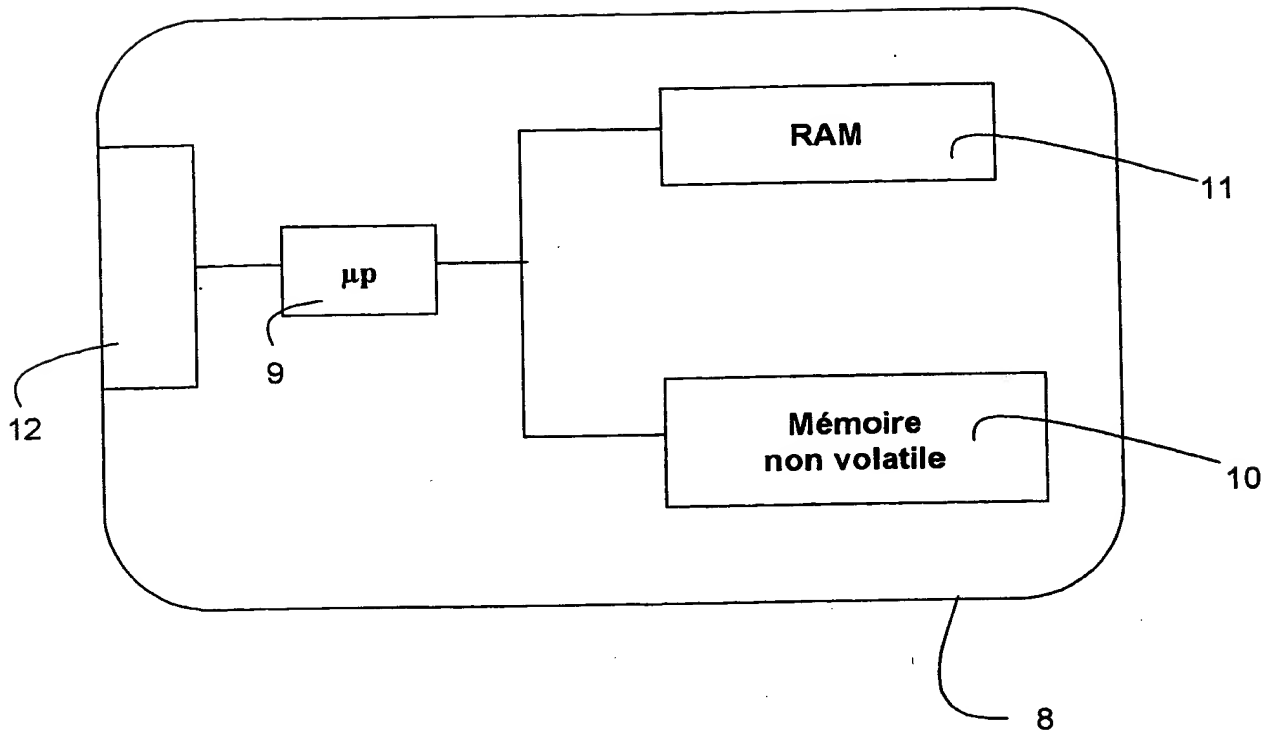

FIGURE UNIQUE